AI-KI-B

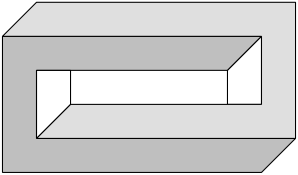
Non-Classical Logics

Ute Schmid & Diedrich Wolter
Practice: Johannes Rabold

Cognitive Systems and Smart Environments
Applied Computer Science, University of Bamberg

last change: 25. Juli 2019, 10:02

Quiz!



1. Alan Turing was recently acknowledged by putting his portrait on the 50 pound note. What is one of his contributions to AI?
2. Is ID3 guaranteed to compute the best classification possible, given the training data?
3. What should happen if you apply the Waltz algorithm for 3D reconstruction to an Escher-like impossible figure?

Topics:

- Natural Language Processing (NLP) overview
- Resolving ambiguity to understand language
- Referring Expression Generation (REG)

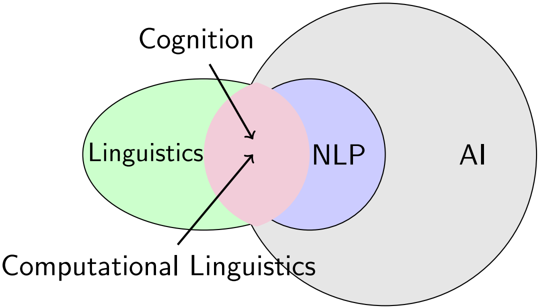
Educational objectives: Being able to...

- explain different tasks tackled within NLP;
- relate NLP tasks to fundamental AI techniques;
- define REG and implement the incremental algorithm (IA).

What is NLP?

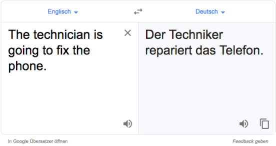
Natural Language Processing (NLP) is sub-area of AI involved with processing speech and language:

- Speech recognition (spoken language)
- Text understanding (written language)
- Language generation (written and spoken language)



(drawing not to scale ;-)

- Automatic translation
- Chatbots, i.e., simple closed-domain question-answering and advertising
- Information extraction from text
- Information retrieval
- (Open-domain) question-answering
- User interfaces



Google Translate

"Go to the third junction and take a right."

```
(do-sequentially
  (do-n-times 3
    (do-sequentially
      (move-to forward-loc)
      (do-until
        (junction current-loc)
        (move-to forward-loc))))
  (turn-right current-loc))
```

example robot instruction (Matuszek et al., 2012)

- Simple program conceived by J. Weizenbaum for experimentation in language-based interaction
- Nevertheless, interaction appears surprisingly natural at first
- Today referred to as basis for chatbots
- User's acceptance of the simple program coined his critical view on AI
- Try yourself, Eliza is included in Emacs: Type Meta-X doctor, to bring it up!

  ELIZA: How do you do? What brings you to see me?
  USER: I hate computers!
  ELIZA: You have your mind on computers, it seems.
  USER: Sure, I hate these things.
  ELIZA: Why do you hate these things?

- Simple program conceived by J. Weizenbaum for experimentation in language-based interaction
- Nevertheless, interaction appears surprisingly natural at first
- Today referred to as basis for chatbots
- User's acceptance of the simple program coined his critical view on AI
- Try yourself, Eliza is included in Emacs: Type Meta-X doctor,

Code snippet from emacs' Eliza shows reply patterns triggered by keywords found in the input (example shown for keyword 'drink(ing)', stored in variable `drnk`):

```
(setq drnk
      '((do you drink a lot of (// found) \?)
        (do you get drunk often \?)
        (($ describe) your drinking habits \.)))
```

Variety of tasks in NLP: **speech recognition**

- Capturing pronunciation, challenged by accents, dialects, emotions, distortion in telecommunication, and noise, especially other speakers
- Identifying and interpreting prosody (intonation, rhythm, stress, and tone)
- Interpreting emotion (e.g., from prosody)

Variety of tasks in NLP: **text understanding**

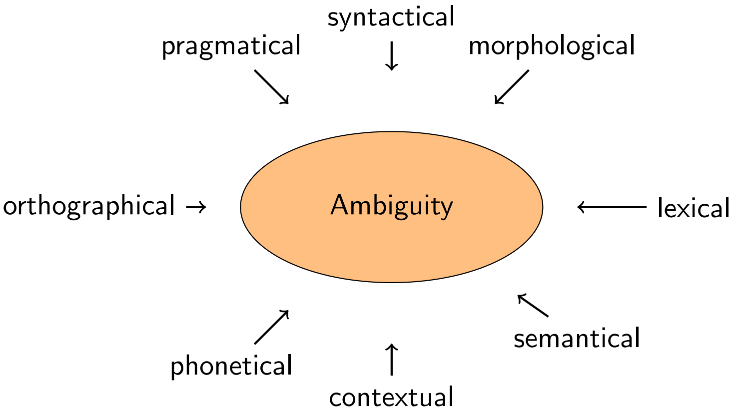
- Developing computational **grammars** for natural language (challenged by complexity, different modes of language use)
- Devising **parsers** robust against grammatical errors (challenged by 'careless speakers')
- Capturing **semantics** of words (AI-Complete![1])
- Capturing context that affects meaning (e.g., in situated interaction)
- Recognising **intent** (e.g., how to answer "Do you know where the Cafeteria is?")
- Recognising named entities (old town hall vs. Old Town Hall)
- Co-reference resolution
- Identifying Causalities

[1] In analogy to NP-complete problems, AI-completeness refers to problems 'any' AI problem can be reduced to

Variety of tasks in NLP: **dialogue and speech production**

- Pre-verbal language production (what to say)
- Verbalisation (how to say it)
- Choosing appropriate intonation and prosody
- Keeping track/exploiting context in conversations
- Deciding when to talk in a dialogue

Challenge in Text Understanding: Ambiguity

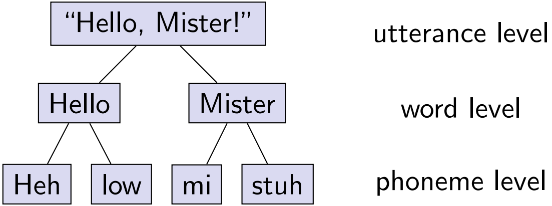


(Wahlster, 2011)

Speech recognition has been shown to benefit from a hierarchical approach combining bottom-up (recognition) and top-down (generative) approaches

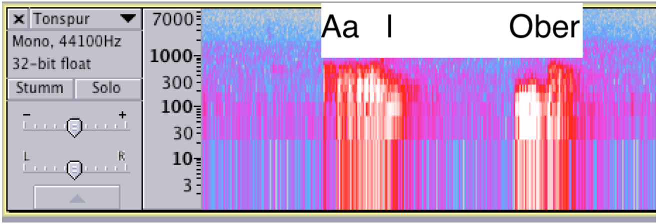
- Starting with an acoustic level: **phonemes**
- Syllable level
- Word level
- Utterance (sentence) level

⇝ Compare to object recognition approaches!



Phonemes are minimal classes of sounds that serve distinct functions in speech acts

- Phonemes can best be analysed in the **frequency domain**, i.e., sound as a function time $\rightarrow$ frequency, not time $\rightarrow$ sound pressure level as recorded in raw audio data
- **Pattern recognition** for identifying phonemes



example made with free audio software Audacity – try yourself!

Problems arising from variations in pronunciation

- Some phonemes may not be uttered/recognised
- Mixup of phonemes occurs

Words are characterised by possible sequences of phonemes

- Similar to finite automata
- But we need to exploit probabilities

Speech recognition is approached as sequence recognition task in a probabilistic model: **Hidden Markov Model (HMM)** ($\rightsquigarrow$ details in SME-PHY-B)

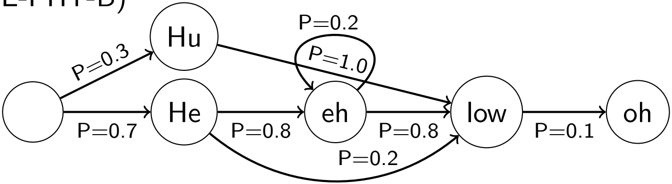


illustration HMM

The **Forward Algorithm** allows us to compute the probability that a word represented as HMM has been uttered, given a probabilistic model of phoneme utterance and their recognition.

Modelling HMMs manually is not feasible

- Machine Learning techniques are thus important
- HMMs can be trained by optimising transitions and recognition probabilities (i.e., parameters of the model) to best fit a set of training data, e.g., using **Baum-Welch algorithm**
- HMMs can also been applied to transition models on the word level, i.e., how sentences are composed out of words
- Lately, logistic regression has also been very successful in recognising words from

HMMs are also predictive models: One can randomly follow transitions in a HMM using the given probabilities for control. Some remarkable approaches in text and music generation have been realised by computing conditional probabilities

$$P(\text{next word} = X | \text{recent words} = Y_1, Y_2, \ldots, Y_n).$$

Fun exercise: Compute conditional probabilities from text (e.g., Shakespeare) and generate random texts. You're in for a surprise (and laugh, occasionally).

> *"For I am shamed by that which I bring forth,*
> *Thy registers and thee I both defy,*
> *Ay me! But yet thou mightst my seat forbear."*
>
> *(anonymous AI)*

Parsing is applied to identify the structure of a sentence using an appropriate **grammar**

- Noam Chomski known from formal languages is a linguist!
- Words are tokens, flexion etc. removed
- Developing suitable grammars is very difficult: grammars should allow semantic structure of input to be mapped to syntactical structures on output (cp. compiler construction)

Example (Tokenising)

"The house, which was recently painted, appears to be for sale."
The/DT house/NN ,/, which/WDT was/VBD recently/RB painted/VBN ,/, appears/VBZ to/TO be/VB for/IN sale/NN ./. "

Try yourself! `http://nlp.stanford.edu:8080/parser/index.jsp`

Parser output as S-Expressions and logic assertions

```
(ROOT
  (S
    (NP
      (NP (DT The) (NN house))
      (, ,)
      (SBAR
        (WHNP (WDT which))
        (S
          (VP (VBD was)
            (VP
              (ADVP (RB recently))
              (VBN painted)))))
      (, ,))
    (VP (VBZ appears)
      (S
        (VP (TO to)
          (VP (VB be)
            (PP (IN for)
              (NP (NN sale)))))))
    (. .)))
}
```

```
det(house-2, The-1)
nsubjpass(painted-7, house-2)
nsubj(appears-9, house-2)
nsubj:xsubj(sale-13, house-2)
ref(house-2, which-4)
auxpass(painted-7, was-5)
advmod(painted-7, recently-6)
acl:relcl(house-2, painted-7)
root(ROOT-0, appears-9)
mark(sale-13, to-10)
cop(sale-13, be-11)
case(sale-13, for-12)
xcomp(appears-9, sale-13)
```

Interesting observation: Words that are synonyms appear in the same context of words.
Idea:

- Fix a corpus (data set) of text, e.g., the whole of English Wikipedia
- Map words to vectors in $\mathbb{R}^N$ for large N, say $500 \ldots 2000$
- Apply optimisation techniques to find a mapping words to vectors that $d(w_1, w_2) \sim d(c_1, c_2)$ where c_i is the lexical context of word w_i
- Choosing 'appropriate' distance functions on word context appears to be crucial, no definitive answer so far.

Observations: synonyms are mapped to close-by locations, semantics of words is (partially) mapped to geometric structures.
Research in NLP and AI is currently exploring this idea of **word embeddings**.

Example: Text Understanding

Human text understanding is often tested by asking questions about text.

"The city councilmen refused the demonstrators a permit because they feared violence."

Who feared violence?

Example: Text Understanding

Human text understanding is often tested by asking questions about text.
"The city councilmen refused the demonstrators a permit because they feared violence."
Who feared violence? Councilmen!

Example: Text Understanding

Human text understanding is often tested by asking questions about text.

"The city councilmen refused the demonstrators a permit because they feared violence."

Who feared violence? Councilmen!

"The city councilmen refused the demonstrators a permit because they advocated violence."

Who advocated violence?

Example: Text Understanding

Human text understanding is often tested by asking questions about text.

"The city councilmen refused the demonstrators a permit because they feared violence."

Who feared violence? Councilmen!

"The city councilmen refused the demonstrators a permit because they advocated violence."

Who advocated violence? Demonstrators!

Example: Text Understanding

Human text understanding is often tested by asking questions about text.

"The city councilmen refused the demonstrators a permit because they feared violence."

Who feared violence? Councilmen!

"The city councilmen refused the demonstrators a permit because they advocated violence."

Who advocated violence? Demonstrators!

Observation: Sentence structure is identical in both examples. Thus we cannot identify what 'they' refers to until we **understand** the meaning of permissions in context of demonstrations. This makes text understanding extremely difficult and thus AI-complete.

In text, many pieces of information required for interpretation are not mentioned explicitly

- Intuitively clear for any human reader or listener
- Requires explicit background knowledge in computers

The **Winograd Schema Challenge** addresses this problem by asking reference resolutions questions that depend on semantics (see Davis and Marcus, 2015)

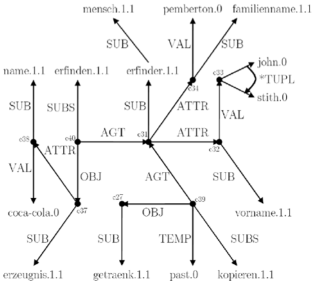
- Change of single word alters interpretation (feared/advocated)
- No statistics or parsing alone can solve the problem
- Examples from collection by Ernie Davis
  - The trophy doesn't fit into the brown suitcase because it's too [small/large]. What is too [small/large]?
  - Paul tried to call George on the phone, but he wasn't [successful/available]. Who was not [successful/available]?

Question-Answering System

The goal is to build a system capable of answering any question asked to it (possibly, googleing the answer or consulting dictionaries, Wikipedia, etc.)

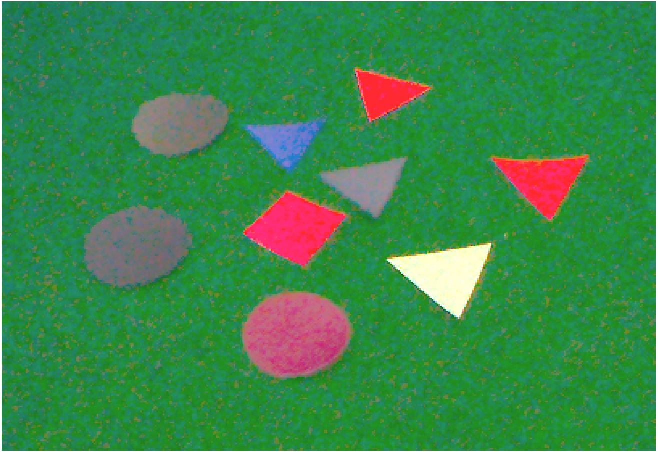
- IBM's Watson was the first successful open-domain QA system, winning Jeopardy against humans
- Systems combine NLP, reasoning and information retrieval techniques
- Today's digital assistants (Alexa, Google, Siri, Cortana) only have very limited QA capabilities

Recall example on semantic networks from KR lecture:



(Furbach et al, 2010)

The semantic network has been automatically extracted from Wikipedia using NLP in the LogAnswer system (Fuhrbach et al., 2010). In LogAnswer, questions are interpreted as logic formulae in a specially designed logic, i.e., NLP connects **text to logic**. Logic reasoning (model checking) can then be applied to compute an asnwer.



Pick a shape!

Referring expressions (REs) are phrases that identify some target object to a listener

- Crucial in situated interaction, e.g., human-robot interaction

perception → object recognition → REG → verbalisation

assumption	Objects in scene have already been recognised
input	Scene description (e.g., list of objects), target object
output	Elements of nominal phrase describing target object
criteria	Intuitive for humans, unequivocal interpretation, robust against variations in

- REs comprise unary descriptions $\mathcal{D}$ (large, green, etc.) and relations $\mathcal{R}$ (left of X, between X and Y)

$\rightsquigarrow$ REG is a recursive task, if relations are considered!

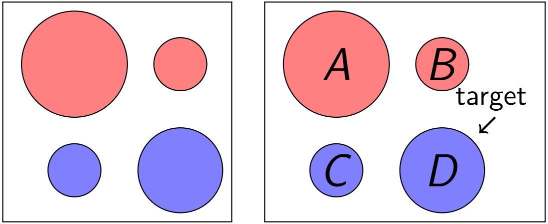
- Assume REs for all objects except for the target object to be known, we can treat relations as sets of unary descriptors
  - example: left_of(T, X), RE(X)=“the cookie” $\rightsquigarrow$ left_of_the_cookie
- REs can thus be modelled as the task of identifying the subset of descriptors RE(X) $\subseteq \mathcal{D}$ that describes target X ‘best’
- We’re involved with a **search problem**!
  - $2^{\mathcal{D}}$ is exponential wrt. to unique descriptions and wrt. amount of objects in scene, if binary relations are considered

Best-first search approach to computing referring expressions

- Popular algorithm proposed by Dale und Reiter (1995) in the sense of multiple revisions, additions, see Kramer & van Deemter (2012)
- Assumes crisp, i.e., Boolean categories: objects are either red or not
- Heuristic to order description candidates is most crucial

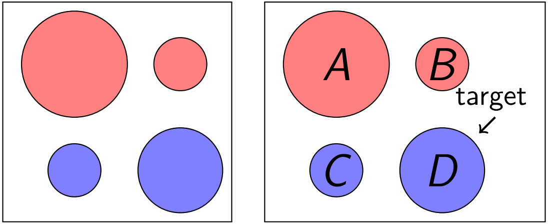
Approach:

- Iterate while current RE does not unequivocally identify target object, i.e., the RE matches to two or more objects
    - Add next description to RE that matches target object



scene — annotated scene

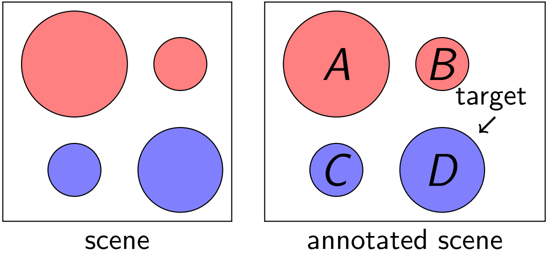
Sequence of attributes: [red, blue, green, small, large]
IA constructs RE E, maintaining a set of ambiguous interpretations D of the uncompleted RE:



scene   annotated scene

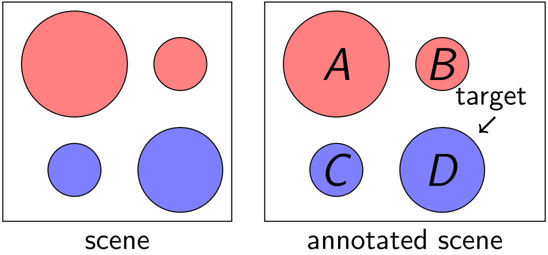
Sequence of attributes: [red, blue, green, small, large]
IA constructs RE E, maintaining a set of ambiguous interpretations D of the uncompleted RE:

1. $E = \{\}, D = \{A, B, C\}$



Sequence of attributes: [red, blue, green, small, large]
IA constructs RE E, maintaining a set of ambiguous interpretations D of the uncompleted RE:

1. $E = \{\}, D = \{A, B, C\}$
2. $E = \{\text{blue}\}, D = \{C\}$



Sequence of attributes: [red, blue, green, small, large]
IA constructs RE E, maintaining a set of ambiguous interpretations D of the uncompleted RE:

1. $E = \{\}, D = \{A, B, C\}$
2. $E = \{\text{blue}\}, D = \{C\}$
3. $E = \{\text{blue, large}\}, D = \{\}$

Incremental Algorithm

function $\mathrm{IA}(O, o_t, A, \pi)$

    $D \leftarrow O \setminus \{o_t\}$ ▷set of distractors

    $L \leftarrow$ **sort**(A, π) ▷list of attributes sorted by heuristic π

    $E \leftarrow \{\}$ ▷referring expression

    for a in list L **do**

        if fits(a, o_t) **then** ▷Boolean class membership test

            $D' \leftarrow D \setminus \{x | x \in D, \neg \textbf{fits}(a, x)\}$

            if $D' \neq D$ **then** ▷rules out some distractors

                $E \leftarrow E \cup \{a\}$

                $D \leftarrow D'$

            end if

            if $D = \emptyset$ **then** ▷no more confusion

                return E

            end if

        end if

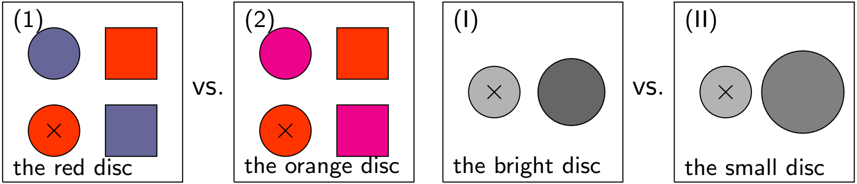
    end for

    return failure ▷no unambiguous expression exists

end function

Need for Refined Linguistic Model

Assigning attributes to objects is not a simple comparison, but requires **context** to be acknowledged:



Influence of context on linguistic encoding in REG: red is more distinguishing in (1), orange is more distinguishing in (2)

Situations where (I) brightness, and (II) size is the most salient property.

Since time for this course is up, we cannot discuss approaches that respect context – if you want to learn more, see for example Mast et al. (2014).

- Natural Language Processing (NLP) comprises:
  - Speech recognition to map spoken language to text
  - Parsing and text understanding
  - Dialogue, Language and Speech Production
- Incremental Algorithm for Computing Referring expressions
- Remarkable mock-ups can be realised with simple techniques (e.g., Eliza)
- Omnipresent challenge: Ambiguity
- Probabilistic models (e.g., Hidden Markov models) are important for computing most likely interpretations
- True text understanding requires knowledge representation and reasoning
- Mastering situated language requires perception, object recognition, and symbolic grounding to be solved

Literature and References

- R. Dale & E. Reiter (1995). Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions, Cognitive Science, 18, 233–263
- E. Davis & G. Marcus (2015). Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence, Communications of the ACM, 58(9), 92–103
- U. Furbach, I. Glöckner, & B. Pelzer (2010). An application of automated reasoning in natural language question answering. AI Commun. 23(2–3): 241–265
- E. Krahmer & K. van Deemter (2012). Computational Generation of Referring Expressions: A Survey, Computational Linguistics, 38(1), 173–218
- V. Mast, Z. Falomir & D. Wolter (2016). Probabilistic reference and grounding with PRAGR for dialogues with robots, Journal of Experimental & Theoretical Artificial Intelligence, 28(5), 889–911
- W. Wahlster (2011). Mit den Dingen sprechen: Autos, Roboter und Weinflaschen als Dialogpartner? In T. Lengauer (ed), Computermodelle in der Wissenschaft – zwischen Analyse, Vorhersage und Suggestion, Nova Acta Leopoldina NF 110, 119–141